## **Client Project Requirement Document**

**Project Title:**  
 **Customer Churn Prediction and Proactive Retention System (CPRS)**

**Deadline: 28th June 2025**

## **🔹 Project Overview**

In today's competitive fintech landscape, **retaining existing customers is significantly more cost-effective than acquiring new ones**. Despite our innovative services, Finexa sees a **monthly churn rate of 4-6%**, directly impacting our revenue and growth targets.

We are initiating a **mission-critical Machine Learning project** to develop a robust, interpretable, and scalable **Churn Prediction and Proactive Retention System (CPRS)**. This system should not only **predict the likelihood of churn** with high accuracy but also **recommend tailored retention actions** that marketing teams can deploy in real time.

This project is a strategic priority and will have **C-suite visibility**. Success in this project could lead to **company-wide deployment and adoption in other regions**.

## **🔹 Project Objectives**

1. Accurately predict customer churn **30-45 days before it happens**.
2. Classify customers into **risk tiers (Low, Medium, High)** with confidence scores.
3. Design **automated rule-based and ML-based retention strategies** for each risk tier.
4. Build a **modular, end-to-end ML pipeline** from ingestion to deployment.
5. Provide **explainable insights** for marketing, product, and support teams.
6. Enable **seamless integration** with our CRM and Campaign Management Systems.
7. Create a **feedback loop** to continuously improve predictions using incoming user data.

## **🔹 Data Requirements**

We will provide data from multiple departments and tools in **CSV, JSON, and API format**, including:

### **1. Customer Profile**

* Customer ID (anonymized)
* Age, Gender, Location
* Occupation, Income Bracket, Credit Score
* Onboarding Date, Channel

### **2. Behavioral Data**

* App and Web Login Logs (timestamps, device type, OS)
* Session Duration, Time Between Sessions
* Feature Usage Logs (loans, savings, card management)

### **3. Transaction History**

* Monthly average balance, debit/credit count
* Loans taken, repayment history, overdraft events
* Declined transactions and failed payments

### **4. Subscription & Plan Data**

* Current plan, plan history, upgrades/downgrades
* Subscription renewal events
* Payment mode (auto/manual), failed payments

### **5. Support & Feedback**

* Support tickets raised (type, resolution time)
* Chat sentiment score (via NLP)
* NPS score, Surveys, Feedback comments

### **6. Label**

* Churned (yes/no), Churn date, Last activity timestamp

We also encourage incorporating **external data sources** like:

* Economic Index (unemployment, inflation)
* Competitor product releases and pricing
* Google Trends / Social Media chatter

## **🔹 Functional Requirements**

### **1. Data Preprocessing & Feature Engineering**

* Detect and handle missing/null/zero values.
* Feature transformation: date-time parsing, session frequency.
* Encoding: Label encoding, one-hot, frequency.
* Aggregation: Rolling averages, monthly statistics, trend slopes.
* Outlier detection (e.g., Isolation Forest).

### **2. Model Development**

* Experiment with:
  + Logistic Regression (baseline)
  + Random Forest, XGBoost, CatBoost, LightGBM
  + LSTM (for time series behavior)
  + TabNet / AutoML tools (e.g., H2O, Google AutoML)
* Train/test split with time-aware cross-validation (if temporal leakage exists).
* Hyperparameter tuning via Optuna or Grid Search.

### **3. Model Evaluation**

* Classification metrics:
  + Accuracy, Precision, Recall, F1
  + AUC-ROC, Log Loss
  + Gini Coefficient
* Business metrics:
  + False Positive Cost (retention offer wasted)
  + False Negative Cost (lost customer)

### **4. Model Interpretability**

* SHAP summary plots and force plots per prediction.
* LIME visualizations for decision reasoning.
* Explain model decisions in business terms (e.g., “Customer is likely to churn because they haven’t used the loan feature in 3 months and have downgraded their plan.”)

### **5. Retention Strategy Engine**

* Assign churned customers to risk segments.
* Recommend offer types: cashback, loyalty points, upgrade prompts, agent calls.
* Allow marketing team to select rule-based or ML-suggested action.

### **6. Deployment & Integration**

* REST API for predictions and explanations.
* Real-time inference capable (fastAPI preferred).
* Dockerize with CI/CD pipeline.
* Deploy to GCP or AWS (preference for Kubernetes or Lambda).
* Schedule retraining (weekly or monthly) with drift detection.

### **7. Visualization Dashboard**

* Customer-level insights: churn probability, risk drivers.
* Segment analytics: churn rates by age, plan, region.
* Time-series trends and retention campaign impact.

## **🔹 Non-Functional Requirements**

|  |  |
| --- | --- |
| **Feature** | **Requirement** |
| **Latency** | < 300ms for a single prediction API call |
| **Security** | Comply with GDPR, anonymize PII data, tokenized access |
| **Scalability** | Handle 1M+ customer records with concurrent inference calls |
| **Monitoring** | Log predictions, monitor data/model drift, raise alerts |
| **Extensibility** | Easy to extend to SME customers and international regions |

## **🔹 Deliverables**

1. Full source code (Python, Jupyter, Flask/FastAPI)
2. Trained model artifacts (.pkl or .onnx)
3. Swagger-based API documentation
4. Dockerfile and Helm charts for deployment
5. Dashboards (Tableau/Power BI/Streamlit)
6. PPT + PDF Report with:
   1. Executive summary
   2. Technical architecture
   3. ML model methodology
   4. Results, visualizations, business recommendations
   5. Limitations and next steps

## **How to Build the Project Without Paid Tools**

### **🔹 1. Data Handling & Storage**

|  |  |  |
| --- | --- | --- |
| **Task** | **Paid Option** | **Free/Open-Source Alternative** |
| Data Storage | AWS S3, Azure Blob | **Local CSVs**, **SQLite**, **PostgreSQL**, **MySQL** |
| ETL & Data Cleaning | Talend, Alteryx | **Pandas**, **Dask**, **Apache Airflow** (for scheduling) |

### **🔹 2. Model Development**

|  |  |  |
| --- | --- | --- |
| **Task** | **Paid Option** | **Free/Open-Source Alternative** |
| AutoML | Google AutoML, H2O.ai Pro | **H2O.ai (open-source)**, **Auto-sklearn**, **TPOT**, **MLJAR** |
| ML Libraries | Azure ML, SageMaker | **Scikit-learn**, **XGBoost**, **LightGBM**, **CatBoost**, **TensorFlow**, **PyTorch** |
| Experiment Tracking | Weights & Biases | **MLflow**, **TensorBoard** |

### **🔹 3. Data Visualization**

|  |  |  |
| --- | --- | --- |
| **Task** | **Paid Option** | **Free/Open-Source Alternative** |
| Dashboarding | Power BI, Tableau | **Plotly Dash**, **Streamlit**, **Grafana**, **Matplotlib**, **Seaborn** |
| Business Insights | Looker | **Streamlit**, **Jupyter Notebook with interactive widgets** |

### **🔹 4. Model Explainability**

|  |  |  |
| --- | --- | --- |
| **Task** | **Paid Option** | **Free/Open-Source Alternative** |
| Model Insights | DataRobot Explainable AI | **SHAP**, **LIME**, **ELI5**, **Skater** |
| Visualization | Commercial AI dashboards | **Plotly**, **Matplotlib**, **Altair** |

### **🔹 5. Deployment**

|  |  |  |
| --- | --- | --- |
| **Task** | **Paid Option** | **Free/Open-Source Alternative** |
| Hosting | AWS Lambda, GCP Cloud Run | **Localhost**, **Render**, **Railway**, **Heroku (free tier)**, **PythonAnywhere**, or **VM on Google Cloud Free Tier** |
| API Framework | Azure Functions, FastAPI Pro | **FastAPI**, **Flask** |
| Containerization | AWS ECS | **Docker (open-source)** |
| CI/CD | GitHub Actions Pro | **GitHub Actions (Free tier)**, **GitLab CI** |

### **🔹 6. Version Control & Collaboration**

* **Git** + **GitHub (free private repos)** or **GitLab**
* **Trello**, **Notion**, or **Google Sheets** for project tracking

### **🔹 7. Churn Retention Logic**

* Use **Python rule engines** or basic **if-else** logic with thresholds
* Customize business rules using **config files (YAML/JSON)**

## **Example Stack (Fully Free/Open Source)**

|  |  |
| --- | --- |
| **Layer** | **Tool** |
| **Data Preprocessing** | Pandas, Scikit-learn, Dask |
| **ML Modeling** | Scikit-learn, XGBoost, CatBoost |
| **AutoML (Optional)** | Auto-sklearn, TPOT |
| **Explainability** | SHAP, LIME |
| **Dashboard** | Streamlit or Plotly Dash |
| **API Deployment** | FastAPI + Uvicorn + Docker |
| **Version Control** | Git + GitHub |
| **Scheduler (optional)** | Apache Airflow (or just cron jobs) |